Beschreibung der Aufgabenstellung

Aufgabenstellung

Die Aufgabenstellung des hier beschriebenen Studienprojektes sieht vor, eine Online-Applikation bzw. Webapplikation zur Anforderungsaufnahme zu entwickeln. Anforderungen sollen mit Hilfe der „Rupp-Schablone“ von Chris Rupp formuliert werden. Die Idee hinter dem Online-Applikationskonzept ist die einfache Möglichkeit des zentralen Managements der Software und der guten Zugänglichkeit für mehrere Anwender auf die gleiche Datenbasis.

Als Hilfestellung zur Bewältigung dieses Projekts liegt eine Kopie des vorangegangenen Studienprojektes vor. Dieses behandelt die Aufnahme von Anforderungen auf Basis der „Rupp-Schablone“ in einer Offline-Applikation und steht zu Analyse des softwareseitigen Aufnahmeprozesses zur Verfügung.

Rupp-Schablone

Die „Rupp-Schablone“ dient der vereinfachten Aufnahme komplexer Anforderungen. Sie nutzt dazu nur eine Teilmenge der natürlichen Sprache. Genutzt wird die Schnittmenge, welche gleichermaßen Effizienz in der Bearbeitung sowie Verständlichkeit für den Anwender darstellt.

<Bild Schnittmenge>

Der Aufbau der Schablone ist an den deutschen Satzbau angepasst. Dies kann mit einer eingeschränkten Form der Subjekt-Prädikat-Objekt Logik betitelt werden. Das erste Element des Aufbaus ist ein Subjekt. Dieses Subjekt stellt den Satzgegenstand dar. Folgend wird die Wichtigkeit der auszuführenden Tätigkeit auf Basis eines helfenden Prädikats festgelegt. Dieses wird von Rupp als rechtlich bindend bezeichnet (auf Vertragsbasis). Das dritte Element ist ein optionales Dativ-Objekt zur Bestimmung des Empfängers bzw. des Nutznießers der Tätigkeit. Mit dem vierten Element wird zwischen einer selbstständigen, oder einer nutzerinitiierten Durchführung unterschieden. An fünfter Stelle steht das vom Subjekt zu behandelnde Objekt und abschließend das eigentlich auszuführende Prädikat. Das Prädikat beschreibt den Umgang bzw. die Beeinflussung des Objekts durch das Subjekt.

<Bild Schablone>

Erweiterungen

Als wünschenswerte Entwicklungen für die Webapplikation seien im Gegensatz zur derzeitigen offline Lösung des vorigen Studienprojekts genannt: ein Organisationssystem zur Zusammenarbeit mehrerer Anwender an einem Anforderungsprojekt, ein Loginsystem zur Authentifizierung der einzelnen Anwender, eine Exportfunktion für die Anforderungen zur Weiterbearbeitung offline und einige Zusatzattribute zu jeder Anforderung (Priorität, Status, Identifikationsnummer, Abhängigkeiten).

Entwicklung

Allgemein

Die Entwicklung der Webapplikation wird in mehreren Schritten durchgeführt. Vorangehend ist eine ganzheitliche Planung und Strukturierung für den erfolgreichen Abschluss des Projekts unabdinglich. Vorausschauend muss die vorgegebene Zeitspanne zur Erfüllung der Aufgabe eingeteilt und unterteilt werden. Es muss ein Konzept und ein Leitfaden entstehen, auf welchem die Entwicklung basiert und durchgeführt wird. Weiterhin müssen Überlegungen zu möglichen Technologien und deren Vor- und Nachteile getätigt werden. Sobald diese Grundbausteine ausgearbeitet sind, ist es wichtig, eine funktionierende Infrastruktur zur Softwareentwicklung zu entwerfen. Innerhalb dieser muss auch die Lauffähigkeit der zu entwickelnden Applikation gewährleistet sein. Damit kann schon jetzt später möglicherweise auftrenden Fehlern vorgebeugt, bzw. diese beschränkt werden. Nach der darauffolgenden Erstellung der Aufgaben werden diese aufgeteilt, um paralleles Arbeiten effizient zu ermöglichen.

Realisierung

Einleitung

Am Anfang der Realisierung steht die Planung sowie die Auswahl der Technologien, Programmiersprachen, Tools bzw. Tool-Chain. Diese sollten sorgsam gewählt werden, da sie die gesamte Entwicklungsdauer über genutzt werden.   
Durch die Analyse der benötigten Architektur und Infrastruktur lassen sich die verfügbaren Tools und Programmiersprachen eingrenzen, welche mit der Umgebung kompatibel sind.

Vision

Damit das Projekt erfolgreich abgeschlossen werden kann, ist es notwendig, sich zuerst eine Idee bzw. Vision des künftigen Produkts zu erschließen. Darunter fallen insbesondere die Funktionalitäten, das Design und die Bedienung. Zu diesen drei Kategorien werden alle Ideen dokumentiert. Durch Revision der einzelnen Vorschläge resultieren nun folgende Ergebnisse:

* Das Tool muss Anforderungen aus Eingabefeldern auslesen können
* Das Tool muss ein Authentifikationssystem bieten
* Das Tool muss ein Teamsystem bieten
* Das Tool muss ein Securitysystem bieten
* Das Tool muss Anforderungen exportieren können
* Das Tool muss das Bearbeiten vorhandener Anforderungen ermöglichen
* Das Design soll Aktualität widerspiegeln
* Das Design soll Simplizität darstellen
* Das Design soll Offenheit implizieren
* Die Bedienung soll intuitiv sein
* Die Bedienung soll einfach gehalten sein

Durch die exakte Umsetzung dieser Punkte sollte das Projekt erfolgreich abschließen. Durch Recherchen und Evaluationen bestimmter Techniken zum Erreichen positiver Nutzererfahrungen soll ein ansprechendes Design erstellt werden, welches den Nutzer durch visuelles Feedback bei der Arbeit unterstützt.

Architektur

Eine Webapplikation zeichnet sich dadurch aus, dass sie im Netzwerk arbeitet. Das bedeutet, dass sie für alle authentifizierten Anwender in diesem Netzwerk zugänglich sein muss. Aus Sicht des Systems ist der Anwender ein Client, welcher Datenpakete an den Server sendet und abfragt. Basierend auf dieser Tatsache wird abgeleitet, dass es hier um eine Client-Server Architektur handelt. Der Server ist deshalb ein Webserver, der die Dokumente für den Client bereitstellt und für Interaktionen mit dem Anwender bereit steht. Da Daten persistent gespeichert werden müssen, so dass über einen längeren Zeitraum mit ihnen gearbeitet werden kann, wird eine Datenbank benötigt. Diese beiden Komponenten (Webserver, Datenbank) sind für die Datenverarbeitung zuständig, während der Client (Browser) die verarbeiteten Daten empfängt und neue Daten / Anfragen an den Server sendet.

Sprachen

Die oben beschriebene vorausgesetzte Umgebung lässt sich am einfachsten mittels XAMPP (Windows, OSX), oder LAMP (Linux) realisieren. XAMPP steht für Apache, MySQL, PHP, Perl und LAMP für Linux, Apache, MySQL, PHP. In beiden dieser Pakete sind jeweils schon ein Webserver (Apache) und eine Datenbank (MySQL, relational) enthalten. Damit die Datenbankanbindung gewährleistet ist, bieten diese Pakete PHP-Unterstützung mit an. Nur so lässt sich die Datenbank ansprechen und es können Daten ausgelesen und abgespeichert werden. Das Frontend wird dementsprechend mittels der Struktursprache HTML geschrieben. HTML erlaubt das Einbetten von (u.A.) PHP Code, falls die PHP-Unterstützung des Webservers aktiviert ist. Die Formatierung der HTML-Struktur wird durch CSS realisiert.

Zur intuitiveren Bedienung der Webapplikation wird zusätzlich das Javascript genutzt. Es erlaubt DOM-Manipulationen (Document Object Model) einfach durchzuführen.

Frameworks

Frameworks stellen schon vorgefertigte Funktionen einer bestimmten Sprache zur Verfügung. In einem größeren Softwareprojekt lässt sich so ohne Mehraufwand, im Vergleich zur puren Programmiersprache, ein funktionales Code-Grundgerüst bauen.

Das Design der Website wird mittels CSS erstellt und setzt auf ein schon strukturiertes HTML-Dokument auf. Derzeit werden allerdings immer mehr Websites über mobile Endgeräte aufgerufen, welche kleinere Displays und/oder Auflösungen als aktuelle Home Computer besitzen. Um das Design trotzdem für alle Geräte zu optimieren, eignet sich CSS nativ nur bedingt. Deshalb wird das CSS-Framework „SASS“ genutzt, welches eine ähnliche Syntax besitzt, jedoch mit weniger Code auskommt. Außerdem wird eine Klassenbibliothek für CSS-Klassen genutzt. Diese heißt „Bootstrap“ und ist in der Webentwicklung weit verbreitet. Bootstrap bietet viele responsive Klasse und Layouts, die z.B. alte nicht-responsive Tabellen-Layouts sehr einfach ersetzen können. Viele Klassen besitzen zusätzliche Effekte bei Hover (Mauszeiger über Element), oder Klickanimationen. Für einige bestimmte Effekte wird weiterhin das von jQuery stammende „jQuery UI“ genutzt, insbesondere für Klickanimationen.

Javascript erlaubt DOM-Manipulationen. Das ermöglicht eine intuitivere Bedienung der grafischen Weboberfläche. Um diese Manipulationen, wie z.B. Fades und Slides einfacher implementieren zu können, bietet das Framework jQuery eine vorgefertigte Funktionsliste, welche nur aufgerufen werden muss.

Tools

Um eine grundlegende Planung durchzuführen, wird das Projektmanagementtool „Trello“ genutzt. Dieses ermöglicht die Erstellung von Story Cards, teambasiertes Arbeiten an sogenannten „Boards“, Zuweisungen von Aufgaben an Teammitglieder und Sortieren aller Aufgaben. Das Tool ist webbasiert, kostenlos und kann unter [www.trello.com](http://www.trello.com) erreicht werden.

Zur Installation der Softwareumgebung wird das XAMPP-Paket unter Windows installiert. Dieses bietet einen Webserver, eine relationale Datenbank und PHP-Anbindung für den Webserver.

Das eigentliche Programmieren erfolgt in „Notepad++“. Ein Texteditor mit Syntax-Highlighting für viele verschiedene Sprachen. Genau diese Funktionalität macht es zu einem einfach nutzbaren und konfigurierbaren Entwicklungstool. Alle in dem Projekt genutzten Sprachen werden vom Highlighting unterstützt.

Zum automatisierten Kompilieren von SASS „.scss“ Dateien in „.css“ Dateien wird das Command-Line Tool „Compass“ genutzt. Dieses besitzt eine eigene Ruby Command-Line, mit welcher die Kompilierung getriggert wird. Es kann außerdem einen Verzeichnispfad beobachten und kompiliert alle dort befindlichen Dateien, sobald eine Änderung der Checksumme eben dieser wahrgenommen wird.

Für Debug-Zwecke dient die Konsole im Firefox-Browser. Über diese können beispielsweise Variablenwerte ausgelesen, Funktionsaufrufe durchgeführt und Elemente untersucht werden.

Entwicklungsprozess

Datenbank

Um die Daten der Applikation zu speichern, wird eine Datenbank aufgesetzt. Diese enthält drei Tabellen. Eine Tabelle für die Nutzerinformationen, eine für die Anforderungsinformationen und eine für die Teaminformationen. Der Zusammenhang dieser besteht mittels Fremdschlüsselvergabe an jeweilige Attribute. Zur eindeutigen Identifikation eines Datensatzes werden Identifikationsnummern vergeben.

Grundstruktur

Nach allen Vorbereitungen werden die manifestierten Ideen umgesetzt. Als Hauptelement dient ein HTML - Template. Das Template ist eigens für diese Applikation erstellt worden und umfasst die Seitenaufteilung in mehrere Bereiche (Layout). Alle Bereiche sind gekapselte Objekte und nicht voneinander abhängig, um eine optimale Modularität zu gewährleisten. Das Template besteht aus Kopf- sowie Fußbereich, aus Inhaltsbereich und aus links positionierter Menüleiste, ist zunächst rein statisch und dient zur Demonstration der Optik.

Für das Authentifizierungssystem wird ein weiteres HTML - Template erstellt, welches als Login-Seite genutzt wird. Auch dieses ist statisch und wird nach und nach im Funktionsumfang erweitert.

Für die Administrationsseite wird eine abgewandelte Form des Applikations-Haupttemplates verwendet, welches nur die notwendigen Funktionen zur Administration enthält, jedoch das gleiche Layout besitzt.

Mittels CSS werden die Seiten formatiert. Die Admin- und Hauptseite greifen auf die selbe CSS-Datei zu, um Einheitlichkeit zu behalten. Für die Loginseite wird eine eigene CSS-Datei erstellt.

Einbau von Funktionalitäten

Loginseite

In die nun fertigen HTML-Seiten werden die Funktionalitäten einzeln eingebaut. Zuerst werden die Authentifizierungsfunktionen implementiert. Dazu zählen:

* Einlesen der Nutzerdaten (Name, Passwort)
* Abgleichen der Nutzerdaten mit den in der Datenbank hinterlegten
* Weiterleitung an die Hauptseite
* Fehlermeldung bei Falscheingabe

Zum Testen des korrekten Logins wird ein Datensatz manuell in die Datenbank eingetragen. Mit diesen eingetragenen Daten wird sich eingeloggt. Um jedem die Möglichkeit zu bieten, das Tool zu nutzen, wird außerdem eine Registrierungsmöglichkeit geboten. Diese erweitert den Funktionsumfang der Seite um:

* Einlesen neuer Daten
* Speicherung der Daten in der Datenbank

Diese Funktionen werden mit dem Registrieren eines neuen Nutzers und dem anschließenden Login mit dessen Daten getestet.

Hauptseite

Die Applikations-Hauptseite muss einen sehr großen Umfang an Funktionalitäten bieten. Grundlegende Funktionen sind hier:

* Aufnahme von Anforderungen
* Darstellung von Anforderungen
* Bearbeiten / Löschen von Anforderungen
* Erstellen von Teams
* Beitreten in Teams / Austreten aus Teams
* Ändern der Profildaten
* Logout aus der Applikation

Zum Testen dieser Funktionen werden Anforderungen erstellt und ausgelesen. Sobald dieser Test erfolgreich ist, wird das Bearbeiten sowie Löschen der Funktionen getestet. Es wird ein Team erstellt, Nutzer diesem Team zugewiesen und das Team gelöscht. Weiterhin werden die Profildaten neu eingegeben und es wird sich aus der Applikation ausgeloggt.

Adminseite

Die Adminseite ist nur dem Administrator zugänglich und bietet einen eingeschränkten Funktionsumfang. Es lassen sich lediglich Nutzer und/oder Teams löschen. Die Nutzerdaten des Admins können außerdem geändert werden.

Zu Testen ist dies, indem ein bestehender Nutzer gelöscht, sowie dessen Team an einen anderen Nutzer übertragen wird.